Embedded Intel® Architecture in Virtual Private Network Design

Embedded Intel® Architecture in Virtual Private Network Design

Contents

Executive Summary

The Internet is quickly becoming the primary vehicle for communication in the business community. The transfer of important and sensitive data over the Internet has become routine. Unfortunately, using the Internet to transfer important data is not as safe as many users may assume. This is because anyone with access to the Internet and the requisite technical skill can intercept data and use the information to cause damage. In addition, the Internet can pose other security risks. A recent study (FBI/CSI Computer Crime & Security Survey, 1999) estimates that financial losses due to computer security exceeded $100 million for the third consecutive year. Obviously, there is a need to provide a safe and reliable way to communicate sensitive data over the Internet.

In the past, organizations that required a secure network were forced to lease public telecommunication lines or use frame relay circuits. This may be a safe solution, but it is also expensive, relatively inflexible, and does not easily support remote users. An alternative solution that is quickly gaining acceptance is the Virtual Private Network (VPN).

The strength of a VPN is its ability to transmit information securely and reliably over the existing unsecured public telecommunication infrastructure. A VPN is a 'virtual network' since connections are established only on an as-needed basis. The transmitted information is encrypted and tunneled point-to-point over a packet-switched unsecure network. At the receiving end, the information is decrypted, filtered if necessary, and checked for integrity. A VPN provides network users with an inexpensive, safe and scalable security solution.

A VPN may be implemented in several ways:

- LAN-to-LAN
- Remote user-to-LAN
- Within an intranet.

Figure 1 illustrates several VPN configurations. The advantages of VPNs have ignited a large interest in the field, and many software solutions exist. However, effectively supporting a VPN demands an abundance of system resources. Ideally, supporting a VPN should not affect network speed or performance. If a system is tasked with other responsibilities, running a VPN on the system will degrade the performance of the network. For example, data encryption and packet routing are processor-intensive tasks, and using a single system to execute both processes will result in reduced network performance.

The solution is to have a dedicated system operate all of the VPN tasks. This dedicated VPN appliance can efficiently manage the process while supporting a superior user experience. To accomplish this, a high-performance processor and system architecture is required in order to execute the required tunneling encryption/decryption algorithm.

Intel provides the optimal system with its embedded Intel® Architecture (EIA) design. A software-based VPN solution that is integrated onto an EIA platform will provide maximum security, scalability and performance. The primary system components of a VPN appliance include the processor, chipset, Ethernet controllers, memory and VPN software.

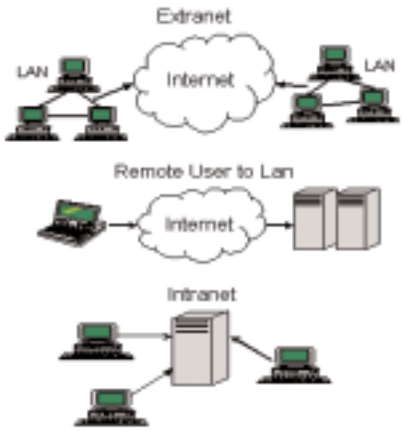

Figure 1 - Typical Virtual Private Network Configurations

Intel Entry-Level VPN Appliance Design

Intel provides the ideal platform to host VPN software. The embedded Intel Architecture (EIA) entry-level design integrates an Intel Celeron™ processor, 440BX chipset, two IEEE 802.3 10/100 Ethernet ports, and memory. The EIA design supports a variety of general-purpose operating systems, including Microsoft Windows* 95/98/2000/NT, Linux, Unix*, and Solaris*. The 440BX chipset ensures that the system will be ready to operate at a 100 MHz system bus speed. Two fast Ethernet ports allow the user to link to a LAN and any other Internet connection device, such as a cable modem, xDSL modem, or server. While the implementation described includes an RS-232 serial port for local console use, a VPN appliance can also be administered remotely over a LAN. Figure 2 shows an entry-level communications design based on embedded Intel Architecture.

The embedded Intel Architecture solution implements a key set of hardware building blocks to address the performance requirements of VPN applications.

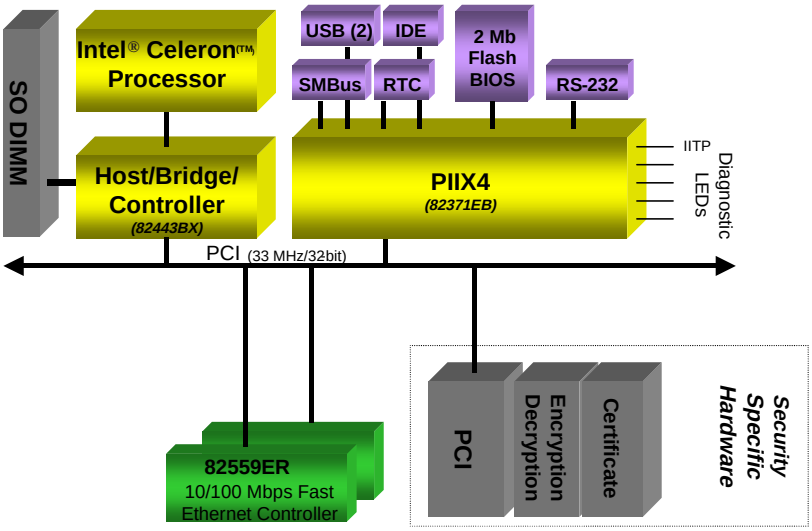

Figure 2 - Reference Configuration Block Diagram

Processor: Intel Celeron Processor

The processor handles encryption/decryption, IP routing and authentication. A 300 MHz processor can handle approximately 1500-2000 tunnel connections. The Intel Celeron processor implements a Dynamic Execution microarchitecture and executes MMX™ instructions for enhanced media and communication performance.

The Celeron processor is based on the high-performance Intel Architecture core. In 300, 366, and 433 MHz versions, the Celeron processor is provided in a Plastic Pin-Grid Array (PPGA) package. Versions of the processor rated at 566 MHz and higher are available in the Flip-Chip Pin Grid Array package. The Celeron processor utilizes the AGTL+ system bus used by the Intel® Pentium® II and Pentium III processors. It includes an integrated 128 Kbyte L2 cache with a separate 16 Kbyte instruction and 16 Kbyte data L1 cache. The L2 cache is capable of caching up to 4 Gbytes of system memory address space.

Intel 440BX chipset

The Intel 440BX chipset improves the speed of the system bus from 66 MHz to 100 MHz, while increasing the width and depth of buffers to the system bus, SDRAM and PCI bus. In addition, the 440BX chipset provides support for ATA/33. The 82443BX has the following features:

- Support for single Celeron processor configuration
- 64-bit GTL+ based host bus interface
- 32-bit host address support
- 64-bit main memory interface with optimized support for SDRAM at 100 and 66/60 MHz
- 32-bit Primary PCI Bus Interface (PCI) with integrated PCI arbiter
- Extensive Data Buffering between all interfaces for high throughput and concurrent operations.

Network Component: Intel® 82559ER Ethernet Controller

The 82559ER Ethernet controller is a fully integrated 10BASE-T/100BASE-TX LAN solution. The 82559ER consists of a Media Access Controller (MAC) and a physical layer (PHY) interface combined into a single component solution that supports 32-bit PCI high-speed data transfer without additional glue logic. Its bus-mastering capabilities enable the component to process high level commands and perform multiple operations, which lowers CPU utilization by off-loading communication tasks from the host CPU.

Operating System, Drivers and Application Software

VPN appliances can be designed to use a variety of operating systems. In this reference design, Linux is selected as the operating system due to its cost, its open configuration model, and a small system footprint.

Characteristics of a VPN Appliance

A VPN appliance provides a simple way to transmit data safely across an unsafe network by creating a 'tunnel' from sender to receiver. The tunnel can be formed from a remote user to a LAN, from LAN-to-LAN, or within an intranet. The connection is set up only on an as-needed basis and broken down when the transaction is complete. This is why the private network is considered virtual. Figure 3 shows a typical VPN.

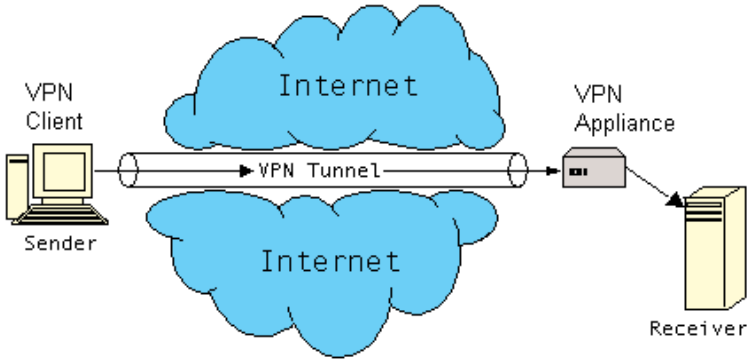

Figure 3 - Virtual Private Network

There are four common protocols for creating VPNs over the Internet:

- Point-to-Point Tunneling Protocol (PPTP)
- Layer 2 Forwarding (L2F)
- Layer 2 Tunneling Protocol (L2TP)
- IP Security Protocol (IPSec).

One reason for the variety of protocols is, for some companies, a VPN is a substitute for remote access servers, allowing mobile users and branch offices to dial into the protected corporate network via their local Internet Service Provider. In other organizations, a VPN may consist of traffic traveling in secure tunnels over the Internet between protected LANs. PPTP has been a widely deployed solution for dial-in VPNs. PPTP builds on the functionality of Point-to-Point Protocol (PPP), the most commonly used protocol for remote access to the Internet. PPTP relies on the authentication mechanisms within PPP, namely Password Authentication Protocol (PAP) and Challenge Handshake Authentication Protocol (CHAP). PPTP is designed to run at Open System Interconnectivity (OSI) Layer 2, the data link layer, while IPSec runs at Layer 3, the network layer. See Figure 4 for OSI model layers. PPTP allows users to create a point-to-point tunnel for transmission security. It also provides multi-protocol support. However, PPTP uses RSA RC-4 encryption, which is not as strong as DES.

L2F tunnels traffic from users to their corporate sites, and it is able to work directly with other media and allows tunnels to support more than one connection. However, L2F does not provide strong encryption and tunnel flow control. L2TP is a Layer 2 protocol and defines its own tunneling protocol. It uses IPSecs for encryption, but leaves the "last mile" unsecured.

Due to IPSecs increasing popularity, it has become the most important protocol out of the four mentioned. By adding security to the IP protocol, IPSec enables safe communication in a TCP/IP connection. It supports several implementations, including remote user dial-in, extranet, and intranet connectivity. In addition, IPSec takes advantage of the latest technologies in encryption and authentication. For example, IPSec uses the Triple Data Encryption Standard (3DES) scheme to provide the privacy needed when transferring data across an unsecure network. IPSec supports Internet Key Exchange (IKE), which authenticates each user and handles key exchange. It is also compatible with X.509 certificate management. IPSec has the advantage of transparency to the user, enabling a secure network without the requirement to reconfigure each system.

(7) Application Layer
(6) Presentation Layer
(5) Session Layer
(4) Transport Layer
(3) Network Layer
(2) Data Link Layer
(1) Physical Layer

Figure 4 — OSI Model

Security and Processor Performance

An abundance of VPN software has been developed, but running this software on a system already tasked with other responsibilities can burden network performance. Because network traffic is bursty, the CPU may be required to perform several functions at once. Encrypting data requires the majority of the processor's resources. If a system is tasked with routing packets, encrypting data, and other server functions, network performance will be compromised. The performance of the network should not be affected by additional security software. By dedicating an independent processing unit to run the VPN software, users can run a secure network without compromising performance. A firewall appliance can be designed to meet the above requirements.

Firewall

The firewall provides access control and prevents unauthorized access to or from private networks, especially intranets. All messages entering or leaving the intranet pass through the firewall, which examines each message and blocks messages that do not meet the specified security criteria. This is one of the fundamental components of a secure network. Firewall functions can include:

- The firewall examines each packet entering or leaving the network and accepts or rejects it based on a predetermined list of criteria.
- The firewall provides an applications gateway that can apply security mechanisms to applications such a FTP or Telnet servers.
- Circuit-level gateways apply security when a Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) connection is established.
- A proxy server intercepts all messages entering and leaving the network, while hiding the true network address.

Like routers, firewalls must process all IP traffic, passing information based on filters that are defined for the firewall. Firewall VPNs are best used when frequent reconfiguration is not required.

Security Issues

Once data has been routed onto the Internet, it is impossible to prevent it from being intercepted (Figure 5 shows an unsecure public network). This does not mean that users cannot implement a secure network. With adequate security methods, users can ensure their data remains private and also determine whether the transmitted data has been compromised. Network security must be evenly distributed, otherwise the network cannot be considered secure. Just as with a linked-chain, the security is only as strong as the weakest link.

Providing proper network security involves more than just data encryption. Many requirements must be satisfied to operate a network efficiently. To enhance network security, the following requirements must be satisfied:

- Authentication
- Privacy
- Data Integrity
- Access Control
- Non-Repudiation
- Network Performance

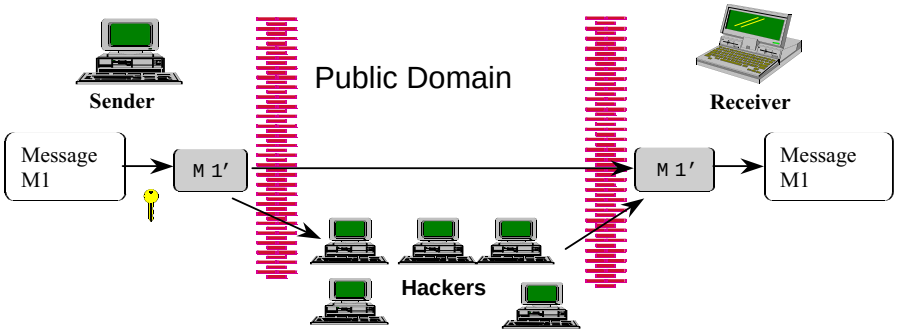

Figure 5 - Unsecure Public Network

Authentication

Authentication is the process of validating the identity of the sender. Many methods exist to perform this function. Password Authentication Procedure (PAP) has the requestor send a username or password. The server will either grant or deny access, based upon whether the username/password is acceptable. Another authentication protocol is Challenge-Handshake Authentication Protocol (CHAP). CHAP has the server send a challenge to the client, the client responds with a one-way hash function, the response is checked for correctness, and access is either granted or denied.

Dial-in authentication methods are also used for remote users. Remote Authentication Dial-In User Service (RADIUS) is used to check the validity of a dial-in user from a remote site. X.509 digital certificates are used to authenticate by establishing a users credentials. A global list of users and their information are stored and may be accessed by authorized users. X.509 v3 is used to pass public keys in IPSec (discussed later).

Privacy

Privacy means that unauthorized users cannot read intercepted data. Privacy is accomplished by encryption and encapsulation. Encryption is the process of scrambling readable data into unreadable data by the use of an encryption key. The scrambled data may only be recovered if the correct key is used to de-scramble it. Encryption makes deciphering the scrambled data mathematically highly improbable and can prevent unauthorized users from accessing the encrypted data.

Many different algorithms are used to encrypt data. The Data Encryption Standard (DES) takes a 64-bit block of data and a 56-bit key and produces a 64-bit block of encrypted data. Reversing the process, using the same key, will retrieve the original 64-bit block of data. Since the key is 56-bit, there are 2^{56} or 7.2×10^{16} different key combinations. 3DES is an even more highly sophisticated encryption procedure, and it provides a much higher level of security. Three keys are used to completely encrypt the data. The original 64-bit block of data is encrypted with Key_1, then decrypted with Key_2, and finally encrypted again with Key_3. The effective key length is 3*56-bit or 168-bits and there are 2^{168} or 3.7×10^{50} different key combinations. Figure 6 maps the 3DES process.

3DES is a symmetric encryption algorithm, meaning that both parties use the same key to encrypt and decrypt. This requires that a key exchange must take place. The Diffie-Hellman protocol allows both parties to acquire the same session key without ever transmitting the full key over the network. Each party chooses half of the session key, and derives the parameters to calculate the second half of the key. Diffie-Hellman uses the Internet Key Exchange (IKE) protocol. IKE handles the exchange of session keys. Each user will have the complete key, without passing the entire key over an unsecured network.

Encapsulation of data is the process of hiding a complete data structure inside another data structure, thereby hiding the inner data structure from the rest of the world. If the network is the Internet, the outer data structure is typically an IP packet. Figure 7 shows a simplified example of an encapsulated data structure.

Encapsulation coupled with encryption yields the privacy needed to safely transmit important information across an unsafe network.

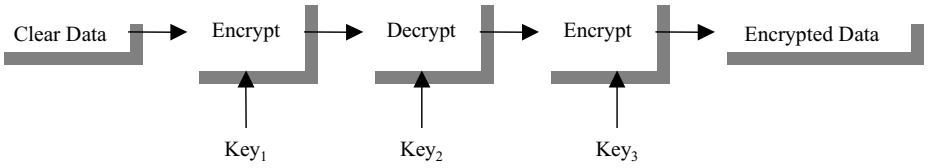

Figure 6 - 3DES Block Diagram

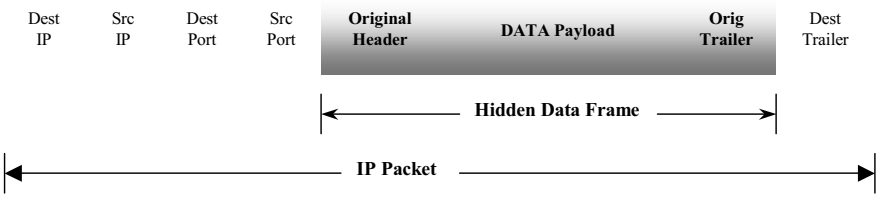

Figure 7 - Encapusalated Data Frame

Data Integrity

Data integrity ensures that the received data has not been altered in-transit. Many protocols have been developed to detect errors after a transmission. One method is to add a parity bit to the end of a packet. The sum of the set bits should equal an even number when even parity is used. Unfortunately, this will only detect an odd number of errors. A more popular method of error detection is Cyclic Redundancy Checking (CRC). This protocol maps a data word into a larger codeword by using a codeword generator. The data link layer and the transport layer of the OSI model play a major role in error detection.

Access Control

Access control is the ability to limit the amount and type of traffic entering or leaving a network. A few examples of access control are:

- Filtering data coming into a network i.e. only allow packets from a specified address
- Filtering data leaving a network. Limiting users to viewing certain Web content (i.e. FTP and HTTP)
- Giving priority to specified data types
- Limiting each users bandwidth to reduce network congestion.

Access control is an essential tool when streamlining any network.

Non-Repudiation

Non-repudiation is the ability of the sender to prove to a third party that a message was sent, or the ability of the receiver to prove to a third party that the exact same message was received. This task may sound trivial, but is actually quite difficult. Many functions must be performed to obtain non-repudiation. The server must authenticate and identify all parties involved in the transaction, and these parties must be approved to perform their desired actions. All of the data transferred must remain intact, and only authorized clients are allowed to access the data.

Network Performance

Network performance can be optimized through the provision of a dedicated system to perform processor-intensive VPN tasks. VPN appliances are designed to efficiently manage heavy application loads, such as the processing requirements of cryptographic algorithms, while supporting high levels of network throughput. The embedded Intel Architecture design provides an ideal platform solution. A software-based VPN application running on an EIA platform optimizes security, scalability and performance.

Intel® Internet Exchange Architecture

The Intel® Internet Exchange architecture (IXA), as shown in Figure 8, provides a consistent framework for OEMs and independent software vendors to quickly deploy new networking and communications services and develop differentiated networking products that deliver scalable performance with reduced total cost of ownership. Intel IXA includes end-to-end development solutions and building blocks that enable developers to create solutions for the entire OSI stack.

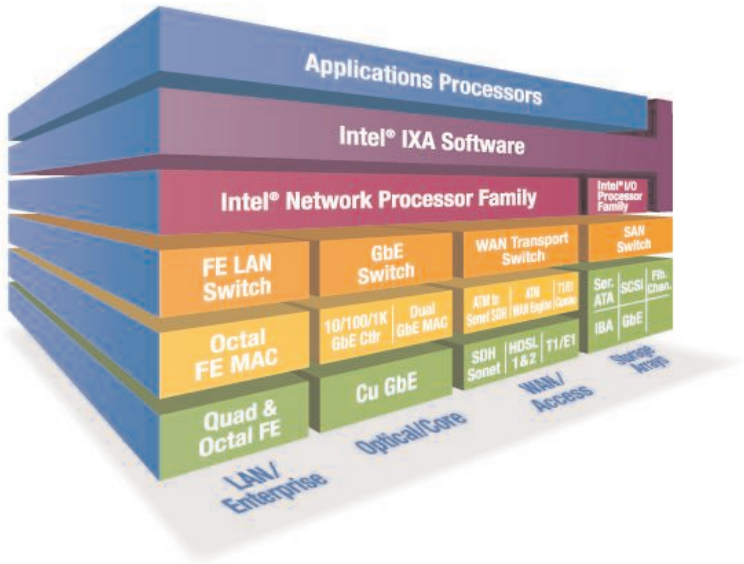

Figure 8 - Intel® Internet Exchange Architecture

Embedded Intel Architecture delivers solutions that meet the performance requirements of the Application Services Layer of the OSI Model. By incorporating scalable embedded Intel Architecture components and software within Intel IXA, Intel is delivering a flexible top-to-bottom architecture that delivers high performance, scalability, code compatibility and programmability that enables faster and more cost effective software-based product differentiation.

Conclusion

The growth of the Internet has sparked a concern for greater network security. To meet this requirement, an abundance of VPN software has been developed. Internet-based VPNs enhance the price and performance of enterprise networks, extranets, and wide-area intranets. VPNs rely on CPU performance for encryption/decryption algorithms, VPN management, user software, and for simultaneous tunnel connections.

Unfortunately, running this software on a system already tasked with other responsibilities can burden the system, and network performance will suffer as a result. The CPU may be required to perform multiple functions at one time. Because processor resources are limited, the CPU cannot route packets, perform server functions, encrypt data, and authenticate users simultaneously without degrading performance.

By dedicating an independent system to run the VPN software, users can maintain a secure network without compromising performance. The features and performance of embedded Intel Architecture systems based on the Celeron processor make such platforms ideal for VPN systems.

For More Information

For more information on Intel's Entry-Level Reference Design for Communications, see:	developer.intel.com/platforms/applied/comm/entry.htm
For additional information on the Intel Celeron processor, see:	developer.intel.com/design/celeron/datashts/243658.htm
For additional information on the Intel 440BX chipset, see:	developer.intel.com/design/chipsets/440bx/index.htm
Details on the Intel 82559 Ethernet controller are available at:	developer.intel.com/design/network/82559.htm
To jump-start the LAN interface in VPN appliance designs, Intel offers a free Linux driver for the 82559 Ethernet controller on its Web site. See Intel's Software Assistant at:	amber.intel.com/scripts-qcube/software/softgridb.asp
For more information on Intel Internet Exchange Architecture Solutions, see:	www.intel.com/ixa

Order Number: 273386-001 Printed in USA/1000/5K/IL2110D LY